CSE 16 7-7-20

Recall:

$$\sim \forall x\, P(x) = \exists x \sim P(x)$$

$$\sim \exists x\, P(x) = \forall x \sim P(x)$$

summarize:

stmt	true when	false when
$\forall x\, P(x)$	$P(x)$ is true for all $x \in U$	$P(x)$ is false for at least one $x \in U$
$\exists x\, P(x)$	$P(x)$ is true for at least one $x \in U$	$P(x)$ is false for all $x \in U$

Ex. Recall

'no king of England is not the king of France'

$$\sim \exists x \left(E(x) \wedge \sim F(x) \right)$$

$$= \forall x \sim \left(E(x) \wedge \sim F(x) \right)$$

$$= \forall x \left(\sim E(x) \vee \sim\sim F(x) \right)$$

$$= \forall x \left(\sim E(x) \vee F(x) \right)$$

$$= \forall x \left(E(x) \Rightarrow F(x) \right)$$

(2.11) Inference Rules

Each inference rule is of the form

$$\frac{\begin{array}{c}\text{hypothesis}\\ \text{(hypothesis)}\end{array}}{\therefore\ \text{conclusion}}$$

all based on tautologies

Modus Ponens

$$\frac{\begin{array}{c}P \Rightarrow Q\\ P\end{array}}{Q}$$

tautology: $[(P \Rightarrow Q) \wedge P] \Rightarrow Q$

Modus Tollens

$$\begin{array}{c} P \Rightarrow Q \\ \sim Q \\ \hline \sim P \end{array}$$

tautology: $[(P \Rightarrow Q) \wedge \sim Q] \Rightarrow \sim P$

Elimination

$$\begin{array}{c} P \vee Q \\ \sim P \\ \hline Q \end{array}$$

tautology: $[(P \vee Q) \wedge \sim P] \Rightarrow Q$

Conjunction

$$\frac{\begin{array}{c} P \\ Q \end{array}}{P \land Q}$$

tautology: $(P \land Q) \Rightarrow (P \land Q)$

sp. case of: $R \Rightarrow R$

Addition

$$\frac{P}{P \lor Q}$$

tautology: $P \Rightarrow (P \lor Q)$

simplification

$$\frac{P \wedge Q}{P}$$

tautology: $(P \wedge Q) \Rightarrow P$

Chapter 3: Counting (combinatorics)

(3.1) Lists

Defn

A list (or sequence) is an ordered collection of objects, where repetition is allowed.

notation: $(1, 5, 2, 2, 1)$

empty list: $(\,)$

objects in a list a called entries, elements or terms.

The number of elements is its length.

Defn

A <u>string</u> is a sequence of symbols from some alphabet, written without commas or parentheses.

<u>Ex.</u> list : $(1, 3, 2, 2, 1)$ length $= 5$

string : 13221

$\{1, 2, 3\}$

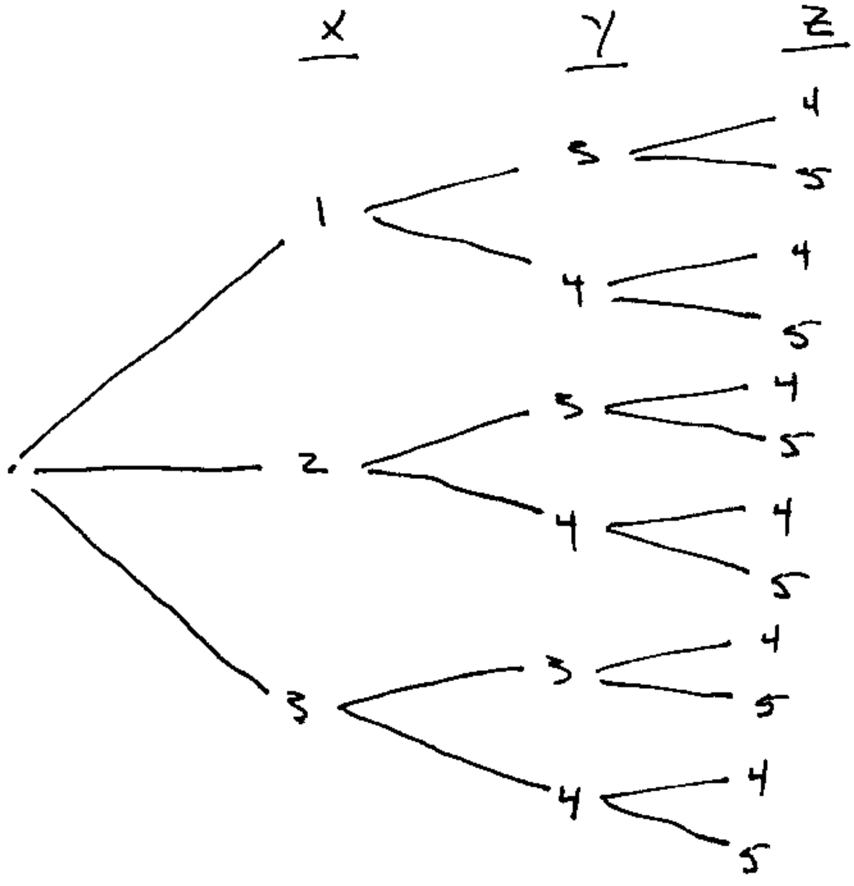
alphabet

<u>note</u>: each symbol must be a contiguous self-contained character, and not ~~itself~~ a string.

(3.2) Multiplication Principle

Ex. find the # of lists (x, y, z) where $x \in \{1, 2, 3\}$, $y \in \{3, 4\}$ and $z \in \{4, 5\}$. Answer: $3 \cdot 2 \cdot 2 = \boxed{12}$

tree diagram



list
(1, 3, 4)
(1, 3, 5)
(1, 4, 4)
(1, 4, 5
(2, 3, 4)
(2, 3, 5)
(2, 4, 4)
(2, 4, 5)
(3, 3, 4)
(3, 3, 5)
(3, 4, 4)
(3, 4, 5)

Generalizing:

$$\left|\{(x,y,z) \mid x \in A \wedge y \in B \wedge z \in C\}\right|$$

$$= |A \times B \times C| = |A| \cdot |B| \cdot |C|$$

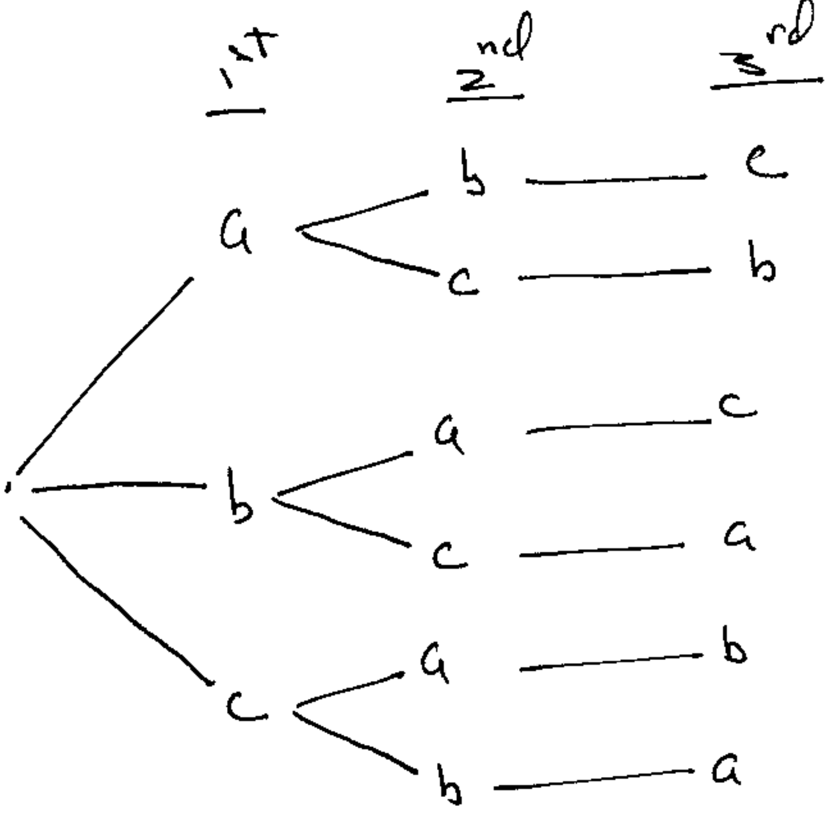
Ex. find the number of strings of length 3 from alphabet $\{a,b,c\}$

$$\underline{1^{st}} \quad \underline{2^{nd}} \quad \underline{3^{rd}}$$

$$\#\text{choices} = 3 \cdot 3 \cdot 3 = 3^3 = \boxed{27}$$

Ex. find # of such strings where no character is repeated.

tree diagram



string

abc
acb
bac
bca
cab
cba

$\# \text{ choices} = 3 \cdot 2 \cdot 1 = \boxed{6}$

Multiplication Principle

Suppose in forming a list $(x_1, x_2, \ldots, x_n)$ there are

a_1 choices for x_1
a_2 " " x_2
$\vdots$
a_i " " x_i
$\vdots$
a_n " " x_n

Then the number of such lists is $a_1 \cdot a_2 \cdot \cdots a_n$

<u>note</u>: in general the # of choices a_i for x_i may depend on the preceding choices

$$(x_1, \ldots, x_{i-1}, x_i, \ldots\ldots)$$

where $2 \le i \le n$.

Ex.

A bit string is a string over alphabet $\{0, 1\}$.

The # of bit strings of length n is : $\boxed{2^n}$

$$\underline{1^{st}} \quad \underline{2^{nd}} \quad \underline{3^{rd}} \quad \cdots\cdots \quad \underline{n^{th}}$$

#choices $= 2 \cdot 2 \cdot 2 \cdot \ \cdots \ \cdot 2 = 2^n$

Recall: $|\mathbb{P}(S)| = 2^{|S|}$

can we use mult. Principle to obtain this?

Yes!

let $|S| = n$, say $S = \{x_1, x_2, \ldots, x_n\}$

To form a subset $A \subseteq S$, we make a seq. of choices

		#
1.	$x_1 \in A$ or $x_1 \notin A$	2
2.	$x_2 \in A$ or $x_2 \notin A$	2
	$\vdots$	$\vdots$
n.	$x_n \in A$ or $x_n \notin A$	2

$$(\#\text{ of sequences of choices}) = \underbrace{2 \cdot 2 \cdots 2}_{n} = 2^n$$

we can actually encode subsets $A \subseteq S$ as bit strings.

$$b_1 \, b_2 \, \ldots\ldots \, b_n$$

where

$$b_i = \begin{cases} 1 & \text{if } x_i \in A \\ 0 & \text{if } x_i \notin A \end{cases}$$

<u>Ex.</u> $S = \{1, 2, 3\}$

$\underline{1}$ $\underline{2}$ $\underline{3}$		
0 0 0	$\longleftrightarrow$	ϕ
0 0 1	$\longleftrightarrow$	$\{3\}$
0 1 0	$\longleftrightarrow$	$\{2\}$
0 1 1	$\longleftrightarrow$	$\{2, 3\}$
1 0 0	$\longleftrightarrow$	$\{1\}$
1 0 1	$\longleftrightarrow$	$\{1, 3\}$
1 1 0	$\longleftrightarrow$	$\{1, 2\}$
1 1 1	$\longleftrightarrow$	$\{1, 2, 3\}$

<u>Ex.</u> The number of strings of length n from an alphabet of size a is : $\boxed{a^n}$

$$\underset{1^{st}}{\_\_} \quad \underset{2^{nd}}{\_\_} \quad \underset{3^{rd}}{\_\_} \cdots\cdots \underset{n^{th}}{\_\_}$$

$$\#\text{choices} = a \cdot a \cdot a \cdots\cdots a = a^n$$

<u>Ex.</u> the # of such strings with no repetitions is

$$\underset{1^{st}}{\_\_} \quad \underset{2^{nd}}{\_\_} \quad \underset{3^{rd}}{\_\_} \cdots\cdots \underset{n^{th}}{\_\_}$$

$$\#\text{choices} = a \cdot (a-1) \cdot (a-2) \cdots\cdots (a-n+1)$$

$$\text{answer} = \boxed{a \cdot (a-1) \cdot (a-2) \cdots (a-n+1) = \frac{a!}{(a-n)!}}$$

(3.3) Addition & Subtraction Principles

Ex.
how many strings of length 4 from $\{a, b, c, d, e, f\}$ are there satisfying

- no repeated letters
- must contain a